Cont C4

an instruction decode mechanism; and

means for said multiplexer choosing input from said first input register.

Ate

31 29. (NEW) The processor of claim 27 further comprising:

32 33 said register decode value having fewer bits than said at least one second source operand.

32 29. (NEW) The processor of claim 27 further comprising;

33 34 a displacement value within said at least one first and said at least one second source operands, said displacement value specifying an effective memory address where data is stored.

33 29. (NEW) The processor of claim 27 further comprising;

34 35 a displacement value within said first and second destination operands, said displacement value specifying an effective memory address where data is stored.

36
34. (NEW) Electromagnetic signals propagating over a computer network, the electromagnetic signals carrying information for practicing the method of claim 9.

REMARKS

This Amendment is filed in response to the Office Action filed January 22, 2002. All objections and rejections are respectfully traversed.

Claims 1-21 are in the case.

Claims 1, 3, 9, 10, 15 and 19 were amended to better claim the invention.

At paragraph 2 of the Office Action it was noted that the IDS had listed pending cases, and that the individual documents were identified only by Attorney Docket Number. Accordingly, a revised 1449 Form is enclosed with the Attorney Docket Number written with the USPTO Serial Number and filing date.

At paragraph 3 of the Office Action the Title was objected to. Accordingly, a new Title was substituted therefore.

At paragraphs 4 and 5 of the Office Action, claims 1-21 are rejected under 35 U.S.C. §102(e) as being anticipated by Asato, U.S. Patent No. 6,145,074. Additionally, various dependent claims were rejected.

The present invention, as set forth in representative claim 1 comprises in part:

> 1. Apparatus for enabling an instruction to control data flow bypassing hardware within a processor of a programmable processing engine, the apparatus comprising:
>
> a pipeline of the processor, the pipeline having a plurality of stages including instruction decode, writeback and execution stages, the execution stage having a plurality of parallel execution units; and
>
> an instruction set of the processor, ***the instruction set defining a register decode value, that specifies one of a first register decode value which defines source operand bypassing, and a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor***.

Asato teaches a data processing device having a bypass function that analyzes the dependency of instructions in a limited range of a processing device with a pipeline bypass function, by making comparisons between destination addresses and source addresses of dif-

ferent instructions. In several embodiments, software is used to determine values of additional fields in the opcode (such as b1, b2, p1 and p2) which determine if bypass of values from other registers along the pipeline would occur. In at least one embodiment, which does not utilize these additional fields in the opcode, dedicated logic is used to determine if the source and destination fields between overlap over a range of consecutive instructions. In all embodiments, the values of these fields or the value produced by the dedicated logic control a subsystem of dedicated logic that determine the input to multiplexers controlling various stages of the pipeline and bypassing between said stages.

Applicants respectfully urge that Asato does not show applicants' claimed novel "***the instruction set defining a register decode value, that specifies one of a first register decode value which defines source operand bypassing, and a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor***". Applicant processor uses bypass determined by using different register decode values for different source operands. If there is dependency, then the code has "***a first register decode value which defines source operand bypassing***" in the case where both instructions, being processed in two different execution units, need access the same source register. The "***first register decode value***" allows the second execution unit to access data from the first execution unit when both instructions. In the alternative case, "***a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor***" indicates the destination of a first instruction is the same register as the source register of the preceding instruction. In either case, the selected register decode values specifies the origin of the data to be read for the value needed for the second instruction.

In contrast, Asato's processor falls under the category of prior art identified in Applicants' specification in which a scoreboard is utilized. Asato compares destination and source addresses of different instructions. In some embodiments, Asato has dedicated software that determines the dependencies between several consecutive instructions, and then determines the values of logical values that determine switches of multiplexers. In sharp contrast, Applicants' invention is concerned with selection of a correct register decode value rather than computing logical values. For this reason, Applicants do not have the aforementioned software or additional logic that Asato teaches, because, as shown in Applicants' specification, they have become unnecessary in Applicants' device.

Applicants respectfully urge that the Asato patent is legally precluded from anticipating the claimed invention under 35 U.S.C. § 102 because of the absence from the Asato patent of Applicants' "***instruction set defining a register decode value, that specifies one of a first register decode value which defines source operand bypassing, and a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor***".

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore are in condition for allowance.

In the event that the Examiner deems personal contact desirable in disposition of this case, the Examiner is invited to call the undersigned attorney at (617) 951-3028.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

A. Sidney Johnston
Reg. No. 29,548
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500

**MARK-UP PAGES FOR THE MAY 21, 2002, AMENDMENT TO
U.S. PATENT APPLICATION SER. NO. 09/390,079**

The replacement for claim 1 resulted from the following changes:

1. Apparatus for enabling an instruction to control data flow bypassing hardware within a processor of a programmable processing engine, the apparatus comprising:

a pipeline of the processor, the pipeline having a plurality of stages including instruction decode, writeback and execution stages, the execution stage having a plurality of parallel execution units; and

an instruction set of the processor, the instruction set defining a register decode value, that specifies one of a first register decode value which defines source operand bypassing, and a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor.

The replacement for claim 3 resulted from the following changes:

3. The apparatus of Claim 2 wherein the register decode value comprises: [one of]

said second register decode value is a result bypass (RRB) operand and said first register decode value is an inter-unit result bypass (RISB) operand, each of which explicitly controls data flow within the pipeline of the processor.

The replacement for claim 9 resulted from the following changes:

9. A method for enabling an instruction to control data flow bypassing hardware within a pipelined processor of a programmable processing engine, the method comprising the steps of:

defining a register decode value that specifies one of a first register decode value which defines source operand bypassing and a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor; and

identifying a pipeline stage register for use as a source operand in an instruction containing the register decode value.

The replacement for claim 10 resulted from the following changes:

10. The method of Claim 9 further comprising the step of explicitly controlling data flow within the pipeline stages of the processor through the use of a register result bypass (RRB) operand in said second register decode value.

The replacement for claim 13 resulted from the following changes:

13. The method of Claim 12 wherein the step of identifying further comprises the steps of:

explicitly specifying the pipeline stage register to be used as the source operand for the instruction. [; and

obviating need to keep track of a scoreboard addressing area.]

The replacement for claim 14 resulted from the following changes:

14. The method of Claim 13 further comprising: [wherein the step of obviating comprises the step of eliminating the need for a scoreboard data structure in the pipelined processor.]

encoding the RRB operand in fewer bits than a regular register operand.

The replacement for claim 15 resulted from the following changes:

15. The method of Claim 14 further comprising:

[the step of] sharing source operand data among the parallel execution units of the pipelined processor through the use of a source bypass (RISB) operand in said first register decode value.

The replacement for claim 19 resulted from the following changes:

19. A computer readable medium containing executable program instructions for enabling an instruction to control data flow bypassing hardware within a pipelined processor of a programmable processing engine, the executable program instructions comprising program instructions for:

defining a register decode value that specifies one of a first register decode value that defines source operand bypassing and a second register decode value that defines result bypassing from a previous instruction executing in pipeline stages of the processor; and

identifying a pipeline stage register for use as a source operand in a current instruction containing the register decode value.